

Realtime Knowledge Management (RKM) – From an International Space Station (ISS) Point of View

Peter I. Robinson (NASA Ames/QSS Group) probinson@mail.arc.nasa.gov
William McDermott (NASA Ames) B.McDermott@nasa.gov
Rick Alena (NASA Ames) Richard.L.Alena@nasa.gov

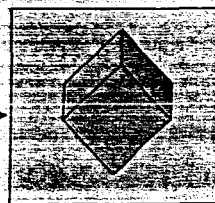
Introduction

When problems occur on spacecraft, onboard and ground-based teams begin to analyze the data. Timely access to relevant domain knowledge can mean the difference between a nominal, degraded or failed mission. For a spacecraft as large as the ISS, the documentation is well over 1 million pages excluding reference materials in the form of spreadsheets, diagrams, databases, source code, test results and other domain-specific forms of knowledge. This is due largely because ISS is being developed in stages, over many years, with many international collaborators. We believe that tools that can provide relevant and timely access to these documents and reference materials will decrease anomaly resolution time with concurrent increases in safety

Diagnostic Data Server:
Serve realtime telemetry, e.g.
Caution and Warning events

Item	Description
1	Low Altimeter Readout
2	Engine Failure - 100%
3	Engine Failure - 100%
4	Engine Failure - 100%
5	Engine Failure - 100%
6	Engine Failure - 100%
7	Engine Failure - 100%
8	Engine Failure - 100%
9	Engine Failure - 100%
10	Engine Failure - 100%
11	Engine Failure - 100%
12	Engine Failure - 100%
13	Engine Failure - 100%
14	Engine Failure - 100%
15	Engine Failure - 100%
16	Engine Failure - 100%
17	Engine Failure - 100%
18	Engine Failure - 100%
19	Engine Failure - 100%
20	Engine Failure - 100%
21	Engine Failure - 100%
22	Engine Failure - 100%
23	Engine Failure - 100%
24	Engine Failure - 100%
25	Engine Failure - 100%
26	Engine Failure - 100%
27	Engine Failure - 100%
28	Engine Failure - 100%
29	Engine Failure - 100%
30	Engine Failure - 100%
31	Engine Failure - 100%
32	Engine Failure - 100%
33	Engine Failure - 100%
34	Engine Failure - 100%
35	Engine Failure - 100%
36	Engine Failure - 100%
37	Engine Failure - 100%
38	Engine Failure - 100%
39	Engine Failure - 100%
40	Engine Failure - 100%
41	Engine Failure - 100%
42	Engine Failure - 100%
43	Engine Failure - 100%
44	Engine Failure - 100%
45	Engine Failure - 100%
46	Engine Failure - 100%
47	Engine Failure - 100%
48	Engine Failure - 100%
49	Engine Failure - 100%
50	Engine Failure - 100%
51	Engine Failure - 100%
52	Engine Failure - 100%
53	Engine Failure - 100%
54	Engine Failure - 100%
55	Engine Failure - 100%
56	Engine Failure - 100%
57	Engine Failure - 100%
58	Engine Failure - 100%
59	Engine Failure - 100%
60	Engine Failure - 100%
61	Engine Failure - 100%
62	Engine Failure - 100%
63	Engine Failure - 100%
64	Engine Failure - 100%
65	Engine Failure - 100%
66	Engine Failure - 100%
67	Engine Failure - 100%
68	Engine Failure - 100%
69	Engine Failure - 100%
70	Engine Failure - 100%
71	Engine Failure - 100%
72	Engine Failure - 100%
73	Engine Failure - 100%
74	Engine Failure - 100%
75	Engine Failure - 100%
76	Engine Failure - 100%
77	Engine Failure - 100%
78	Engine Failure - 100%
79	Engine Failure - 100%
80	Engine Failure - 100%
81	Engine Failure - 100%
82	Engine Failure - 100%
83	Engine Failure - 100%
84	Engine Failure - 100%
85	Engine Failure - 100%
86	Engine Failure - 100%
87	Engine Failure - 100%
88	Engine Failure - 100%
89	Engine Failure - 100%
90	Engine Failure - 100%
91	Engine Failure - 100%
92	Engine Failure - 100%
93	Engine Failure - 100%
94	Engine Failure - 100%
95	Engine Failure - 100%
96	Engine Failure - 100%
97	Engine Failure - 100%
98	Engine Failure - 100%
99	Engine Failure - 100%
100	Engine Failure - 100%

ISStrider:
Serve diagnoses,
recoveries and
document pointers



Netmark:
Serve ISS documents,
e.g. requirements doc.

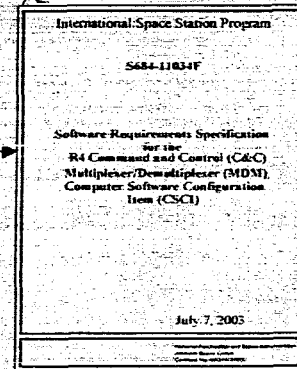


Figure 1: Realtime Knowledge Management Example Realtime telemetry determines diagnostic state which is mapped to documents, e.g. map Caution and Warning event [10] to ISS Requirements documentation.

We are developing automated methods to provide realtime access to spacecraft domain knowledge relevant a spacecraft's current operational state. The method is based upon analyzing state-transition signatures in the telemetry stream. A key insight is that documentation relevant to a specific failure mode or operational state is related to the structure and function of spacecraft systems. This means that diagnostic dependency and state models can provide a roadmap for effective documentation navigation and presentation.

Diagnostic models consume the telemetry and derive a high-level state description of the spacecraft. Each potential spacecraft state description is matched against the predictions

of models that were developed from information found in the pages and sections in the relevant ISS documentation and reference materials. By annotating each model fragment with the domain knowledge sources from which it was derived we can develop a system that automatically selects those documents representing the domain knowledge encapsulated by the models that compute the current spacecraft state. In this manner, when the spacecraft state changes, the relevant documentation context and presentation will also change.

Architecture

The Realtime Knowledge Management (RKM) tool is being developed as an integration of three existing software tools: 1) the Diagnostic Data Server [1], a telemetry server that which provides a temporally organized set of telemetry, logs and data-dumps of the ISS over a selected window of time 2) the Netmark [2] document database that indexes documents both on context (document token) and content (ISS token); and 3) the ISStrider [3] model-based diagnostic tool [3] developed using L2 [6] which models both the hardware and software aspects of the ISS Command and Data Handling (C&DH) system. These three technologies are part of a set of engineering support tools that will be deployed at NASA JSC in the next two years.

The flow of information for the RKM architecture follows the numbers in Figure 1: 1) Telemetry queries are defined by the diagnostic tool ISStrider (or by the human operator) 2) telemetry is consumed by ISStrider, which produces 3) diagnoses (in the future recoveries as well) and 4) document queries. The document queries to Netmark produce 5) relevant documents. The telemetry, diagnoses and relevant documents are all integrated in user-defined GUIs for rapid access by ISS flight engineers.

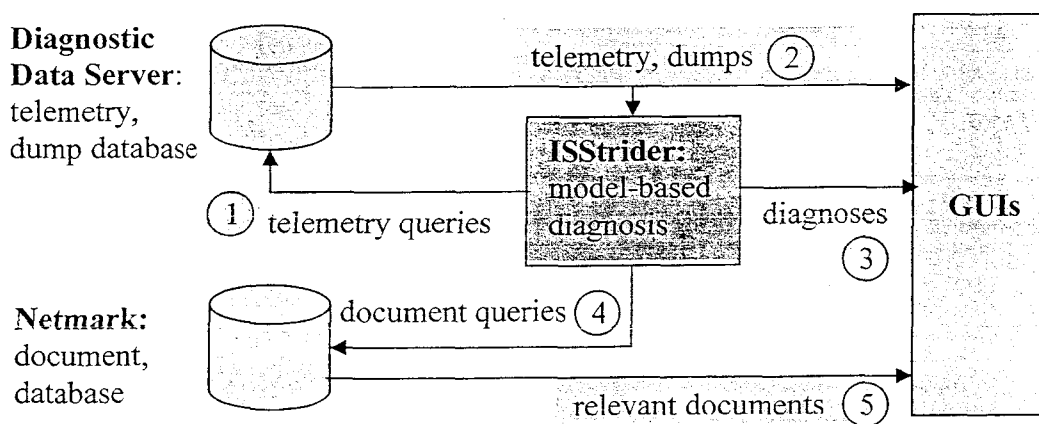


Figure 2: Realtime Knowledge Management (RKM) architecture which integrates DDS [1], Netmark [2] and ISStrider[3].

Example

We demonstrate the RKM architecture in Figure 1. using an example from the ISS C&DH domain. The DDS serves a realtime Caution and Warning event to ISStrider which maps this event to a portion of the ISS software requirements specification (SRS)

[4]. This example defines a direct mapping between telemetry and corresponding documentation (see Implementation Section for details).

In general the mapping from a Caution and Warning event to domain documentation is non-trivial and could require methods that reason about hidden state. For example, the documents relevant to interpret a bus Caution and Warning event (e.g. CW #5392), include a hardware schematic, Bus Address Assignments Table, and the bus profile. Or for example the documents relevant to interpret a computer (MDM) failure Caution and Warning event (e.g. CW #5104) will require all documents related to both the failed execution of the hardware and the software. This would include documents for the hardware and software schematics, for requirements specifications, for documents which define the Built-in Tests (BIT) and Power-On Self-Tests (POST) tests, for documents which define the rate monotonic scheduler and its tasks, and of course for the source code itself [8,9,10,11,12,13,14,15,16,17].

We are developing additional model-based methods based on internal state variables to allow for indexing documents based upon the internal structure and topology of the subsystem domain. Below we define the grammar required to implement the RKM system. It will also apply to state-less dependency models such as TEAMS as well. For TEAMS additional elements will be added to define tests and the source knowledge required to model each test.

Implementation

The implementation is partitioned into three areas, 1) the grammar to describe the ISS document dependencies and the interface into the Netmark document store/database 2) the grammar to describe the ISStrider model-based diagnosis models and its links to the document dependencies. Once the document dependencies are defined, we can annotate our models with the source documents. Our models are based upon the ISStrider system, which utilizes the L2[6] model-based diagnosis system. Our approach is also applicable for many other diagnostic systems including the stateless structural dependency TEAMS[7] tool. For TEAMS focus would be on defining the tests, while for L2 focus is on defining the states of the models and its transitions. This is accomplished in L2 through the use of a model development interface which allows the user to annotate each model fragment with documentation.

Document dependency grammar. To develop a system that can capture ISS specific document types, we define a document dependency grammar. The grammar serves to map the ISS document types to the Netmark tags for both context and content. **Context** is defined as the sections of documents that are important to index into. We have identified a preliminary set of ISS document contexts: 1) document name 2) section 3) page-slide 4) figure 5) table 6) weblink 7) workbook 8) database 9) source DDS dataset. **Content** is defined with respect to ISS domain features. We have identified a preliminary set of ISS document contents: 1) requirement number 2) PUI 3) CSCI (software program identifier) 4) part number 5) procedure number.

`<doc_dependency> ::= <context>+<content>`

```

<context>::=<document_name>|<section>|<page-slide>|<figure>|<table>|
    <web-link>|<workbook>|<database-table>|<DDS dataset>|...
<content>::=<requirement number>+|<PUI>+|<CSCI>+|<Part Number>+|
    <procedure number>+|<CSCI>+

```

Model-based diagnosis grammar. An ISStrider Livingstone model is made up of a set of components which are connected via a set of connections. At each level, including the model level, we can annotate the model fragment with documentation dependencies which map model constructs to source documents.

```

<model>::=<component>+ (<connection>)* <doc_dependency>*

```

Each component is defined at its boundaries by a finite set of ports, each of domain specific types. Internal to each component, is a finite-state machine which itself is made up of a finite set of states. Directed and undirected transitions between the states are defined. Executing the transitions in the forward direction is used to perform state estimation, while executing the state transitions in the reverse direction is used to perform regulation (recovery).

```

<component>::=<port>+ <finite_state_machine> <doc_dependency>*
<finite_state_machine>::=<state>+ <transition>* <doc_dependency>*
<transition>::=<guard> <cost> <state> <state> <transition_type><doc_dependency>*
<state>::=<logical expression> <doc_dependency>*

```

Each connection between components is defined with a type, portname and connection specific documentation dependencies. Each port is defined by a type that identifies the port as an observation, command or internal port as well the possible values that the port can take. Each value is an element of the set of all possible values: $\{..v_i, v_j.. \}$.

```

<connection> ::= <type> <port> <doc_dependency>*.
<port> ::= <port-type> <value-type> <doc_dependency>
<port-type> ::= [observation | command | internal] <doc_dependency>*
<value-type> ::= [(discrete|continuous) <value>+ ] <doc_dependency>*
<value> ::= {..v_i, v_j..} <dependency>*

```

Discussion

At any point in time, a ranked set of diagnoses from the ISStrider system exists. Each of these diagnoses provide a state vector over the state of each component in the system. By identifying the active model fragments, the RKM system, given the current diagnosis state, can automatically present the user with the documentation entailed by the diagnosis. Since the documentation is being driven by the telemetry, often it will occur that many sets of documents are available. This is due to the fact that each diagnosis is composed of a set of components and connections. For each components and connections different doc_dependencies exist which could point to multiple document sources. We would like to compute an intersection of the raw documents in order to provide a unified view to the user. For example, if two model fragments point to two requirements (x.x.x.x and

y.y.y.y). We can first perform a **context search** on requirement(x.x.x.x) in Netmark. which can return a set of pages over a range [lb₁..ub₁] in a ISS requirements doc, (e.g [4,11]). We can then further constrain this page range through a **content search** on requirement(y.y.y.y) over the range of pages [lb₁..ub₁]. This will return a more refined set of pages [lb₂ .. ub₂] such that lb₁ <= lb₂ & ub₂ <= ub₁.

References

- [1] Fletcher, D. P., Alena, R. "A Scalable, Out-of-Band Diagnostics Architecture for International Space Station Systems Support", IEEE Aero 2003
- [2] Maluf, David A. and Tran, Peter B., "NETMARK: Adding Hierarchical Object to Relational Databases", Intelligent Systems Design and Applications (ISDA) 2003, Tulsa, Oklahoma, Conference Proceedings, 2003.
- [3] P. Robinson, M. Shirley, D. Fletcher, R. Alena, D. Duncavage, C. Lee, "Applying Model-Based Reasoning to the FDIR of the Command & Data Handling Subsystem of the International Space Station," iSAIRAS 03 2003
- [4] S684-11034F Boeing ISS Program "Software Requirements Specification for the R4 Command and Control (C&C) Multiplexor/Dimultiplexors (MDM) Computer Software Configuration Item (CSCI)", July 2003
- [5] Schumann, J. Robinson P., [] or SUCCESS is Not Enough: Current Technology and Future Directions in Proof Presentation , Workshop on "Future Directions in Deduction", International Joint Conference of Automated Reasoning IJCAR 2001
- [6] Kurien, J. , Nayak. P.P Back to the Future for Consistency-based Trajectory Tracking. AAAI-97
- [7] Deb, S., Domagala, C., Ghosal, S., Patterson-Hine A., Alena , R. Remote Diagnosis of the International Space Station utilizing Telemetry Data SPIE April 2001
- [8] D684-10500-04B CDH ADD Vols. 1-4.
- [9] C&W Fault Trees: <http://hsi.jsc.nasa.gov/Cwad/>
- [10] Owens, D. , Dempsey, R. Caution and Warning Systems Brief NASA JSC DF25 CDH 06/02
- [11] S684-11032 ISS Software Requirements: R2,, Command and Control (C&C) MDM (CSCI) 11/01
- [12] D684-11106-01 ISS Command and Control Software , Software User's Manual R2 10/01
- [13] D684-10056-01K ISS Prime Contractor Software Standards and Procedures Specification 12/00
- [14] D684-10177-01F ISS Mission Build Facility Standard Output Definition
- [15] NAS15-10000 Software Rqmts Spec for the MDM Boot and Diagnostics Firmware of C&DH
- [16] D684-11111-01 ISS Command and Control Software, Software Top Level Design Document R3
- [17] JSC 28721 User's Guide for the Portable Computer System (PCS) ISS MOD 5/2001
- [18] D684-11379-01 MADE Design Document 3/02

- [20] Sowizral, H., Rushforth, K., Deering, M. "The Java 3D API Specification, 2nd Ed" The Java Series Addison-Wesley 2000
- [21] Mark, H., Levine, A. "The Management of Research Institutions – A Look at Government Laboratories" NASA SP-487
- [22] Maluf, David A. and Tran, Peter B., "NETMARK: Adding Hierarchical Object to Relational Databases", Intelligent Systems Design and Applications (ISDA) 2003, Tulsa, Oklahoma, Conference Proceedings, 2003.
- [23] ISS ECWA CWAD web pointer
- [24] S684-11034F Boeing ISS Program "Software Requirements Specification for the R4 Command and Control (C&C) Multiplexor/Dimultiplexors (MDM) Computer Software Configuration Item (CSCI)", July 2003
- [25] Robinson, P., Mc Dermott, B., Alena R. Realtime Knowledge Management (RTM) – from an ISS point of view, Engineering for Complex Systems (ECS) Iron Bird Conference March 2004
- [26] McRae, G., Los Angeles Times Caltech 1979
- [27] Converse, P. "Religion and Politics: The 1960 Election" in Elections and the Political Order Authors: Campbell, Converse, Miller and Stokes
- [28] 3D Strip chart software <http://www.softwarefx.com/ChartFX/CfxProducts>
- [29] Owens, D., Dempsey, R. *Caution and Warning Systems* Brief NASA JSC DF25 CDH 06/02